

1

《C/C++ 学习指南》

第10.3讲：结构体的深入认识

作者：邵发 QQ群：417024631
官网：http://www.afanihao.cn/c_guide/
答疑：<http://www.afanihao.cn/kbase/>

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

2

版权所有，侵权必究

内容提要

1. 结构体的大小：对齐问题
2. 结构体变量作为成员
3. 结构体作为参数：再论“传值与传地址”的效率问题

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

结构体的大小：对齐问题

结构体的大小跟成员有关，但有时会比各成员之和要大

```
struct Object
{
    char a;
    int b;
};

printf(“%d \n”, sizeof(Object)); // ??? 不是1+4
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

结构体的大小：对齐问题

为了实现对齐，编译器会对结构体进行填充Padding。。。。
(注：C/C++标准没有规定一定要对齐，这是编译器自己的行为)

例如，通常的对齐规则为：

short：其内存地址必须是2字节对齐（地址能被2整除）

int ：4字节对齐

为什么要对齐：因为cpu的指令要求，只能在对齐的地址上存取，属于指令集的语法要求。不对齐就没法存取。

（并不是所有的cpu都有此限制）

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

5

结构体的大小：对齐问题

如何取消对齐规则？使其没有填充？

- ：不同的编译器都不同的方法, 这不在C++标准中定义
- ：但是：永远不要这么做！！

结构体对齐是十分必要的机制，没有任何理由要取消对齐。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

6

版权所有，侵权必究

结构体作为成员

struct的成员的类型可以是另外一种struct类型

```
struct Score
{
    float chinese;
    float english;
    float math;
};
struct Student
{
    int id;
    char name[16];
    Score score; // 此成员变量的类型是struct
};
```

注意前后顺序：把Score类型的定义放在Student的前面。

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

7

结构体作为成员

在内存视图上，仍然很简单，各成员在内存里依次排列
(可能有少许padding)

访问成员的子成员：仍然是使用点号

```
Student s;  
s.score.english = 89.5;
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

8

结构体作为成员

另外一种形式：（这种应用场景相对复杂）

```
struct Student  
{  
    int id;  
    char name[16];  
    Score* ps; // 使用指针，而不是直接包含  
};  
Score _score = { 88.0, 90.0, 98.0 };  
Student _stu;  
_stu.ps = &score;
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

结构体作为参数：再论“传值与传地址”

结构体的体积较大，占用的内存空间较多，往往不使用传值方式。

```
void test(Contact who)
{
    printf("id:%d, name:%s \n", who.id, who.name);
}
```

两大问题：

- (1) 空间因素：使用了更多的内存空间
- (2) 时间因素：将值拷贝到参变量who，要花费较多的cpu

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库

版权所有，侵权必究

结构体作为参数：再论“传值与传引用”

结构体作为函数的参数时，总是用“传地址”方式。如果只是输入参数，则加上const修饰。

```
void test(const Contact* who)
{
    printf("id:%d, name:%s \n", who->id, who->name);
}
```

《C/C++学习指南》 邵发 <http://afanihao.cn> 全套免费教学视频/配套书本/配套题库